

**PROCÉDURE DE RECONNAISSANCE DE L'ÉCRITURE
MANUSCRITE, BASÉE SUR DES CHAINES DE MARKOV CACHÉES,
ET APPLIQUÉE A UN VOCABULAIRE LIMITE**

AVILA M, OLIVIER C., PAQUET T., LECOUTIER Y.

La3i-LACIS, Université de Rouen
Place Emile Blondel, 76821 Mont Saint Aignan Cedex, FRANCE

RÉSUMÉ

ABSTRACT

Une méthode de reconnaissance de l'écriture manuscrite appliquée à un lexique de faible taille est proposée. L'application envisagée concerne le vocabulaire de libellé d'un chèque. L'approche choisie est une approche markovienne qui dans un premier temps est mise en place au niveau des enchaînements de mots et pour laquelle la probabilité de chacun des états est évaluée en utilisant des primitives globales telles que hampes et jambages. Les premiers résultats laissent envisager une extension possible de la méthode au niveau des enchaînements de graphèmes dans le mot afin d'affiner les probabilités d'émission de chacun des mots.

In this paper, we propose a method for recognizing handwritten words from a limited index.

The application in view concerns the vocabulary of cursive words on cheques.

We chose a Markovian approach which is first applied to the sequences of words, and for which the probability of each state is calculated by using global features such as ascenders and descenders.

The first results allow us to consider an extension of the method to the sequences of graphemes in a word in order to improve the probabilities of each word.

INTRODUCTION

Nous présentons les premiers résultats de reconnaissance de phrases manuscrites destinée à une application de lecture automatique de montants de chèques bancaires. L'approche retenue pour la phase de prétraitement des images (binarisation et segmentation de la zone de texte dans l'image) est présentée dans [COURT92].

L'objectif étant la reconnaissance de phrases manuscrites, nous présentons dans une première partie les méthodes développées à ce niveau, et basées sur une description markovienne de l'enchaînement des mots dans la phrase.

Dans une deuxième partie, nous exposons une méthode pour la localisation et l'extraction des informations nécessaires à la mise en place du modèle de Markov sur la phrase [PAQ93].

Dans une troisième partie, nous présentons le module utilisé pour la reconnaissance des phrases, en définissant les états du HMM.

La dernière partie de cet article est consacrée à la présentation des premiers résultats mettant en évidence l'intérêt de la méthode. Un exemple est proposé montrant le fonctionnement de ce module.

I- APERÇU THÉORIQUE

I-1- Modèle de Markov caché (HMM)

Rappelons la définition d'un processus de Markov: Soit un système pouvant être décrit, en toute position k , comme un

ensemble d'états (x_0, x_1, \dots, x_k) choisis parmi n états distincts (e_1, e_2, \dots, e_n) . On suppose qu'en chaque k , le système subit un changement d'états avec possibilité de retour au même état. Le système $(x_0, x_1, \dots, x_k, \dots)$ est alors décrit, s'il s'agit d'une chaîne de Markov d'ordre 1, par l'ensemble des probabilités:

$$P(x_0 = e_i) = \pi_i, \quad i = 1, \dots, n.$$

$$P(x_{k+1} = e_i / x_k = e_j, x_{k-1} = e_l, \dots) = P(x_{k+1} = e_i / x_k = e_j).$$

Si ce dernier terme est indépendant de la position considérée, on obtient une matrice A de probabilité de transition notée (a_{ij}) , $i, j = 1, \dots, n$.

Si x_k ne correspond pas à un événement physiquement observable, on a alors un modèle de Markov caché (HMM), où l'observation est une fonction aléatoire de x_k . Il en résulte un modèle formé de 2 processus aléatoires dont l'un n'est pas observable. Un HMM est caractérisé par 5 paramètres $\Lambda = (n, m, A, B, \Pi)$, n, m, A, Π et B étant définis comme suit (nous reprenons ici l'articulation de [RAB89]) :

- n est le nombre d'état du modèle (e_1, \dots, e_n) .

- m est le nombre d'observations distinctes possibles par état: $z = z_1, \dots, z_m$.

- A est la matrice des probabilités de transition.

- B est la matrice des probabilités des observations relativement aux états: $B = b_{il}(l) = P(z_l \text{ à la position } k / x_k = e_i)$, avec $i = 1, \dots, n; l = 1, \dots, m$.

- Π est le vecteur des états initiaux (π_i) , $i = 1, \dots, n$.



Λ étant connus, le HMM va être utilisé sur une suite $O = o_0, \dots, o_T$ où chaque o_k est une des observations z_k . Notre problème est alors le suivant: étant donné O et Λ , comment choisir x_0, \dots, x_T optimale relativement à O .

I-2- Algorithme de Viterbi

On cherche la suite $x = (x_0, \dots, x_T)$ rendant maximale $p(x/O)$, soit à maximiser:

$$P(x)P(O/x) = \prod_{k=0}^{T-1} P(x_{k+1}/x_k) \cdot \prod_{k=0}^{T-1} P(z_k/x_{k+1}, x_k).$$

Notons ζ_k la transition (x_{k+1}, x_k) . Considérons alors $P(x_{k+1}/x_k) \cdot P(z_k/x_{k+1}, x_k)$ et notons $d(\zeta_k)$ le "poids" de cette transition:

$$d(\zeta_k) = -\ln P(x_{k+1}/x_k) - \ln P(z_k/x_{k+1}, x_k).$$

Le problème énoncé se réduit donc à la minimisation de

$$\sum_{i=0}^{T-1} d(\zeta_i).$$

Notons enfin x_0^k une suite possible d'états commençant à x_0 (de probabilité initiale π_i) et finissant à x_k . Le "poids" de chaque

chaîne x_0^k est donc $d(x_0^k) = \sum_{i=0}^{k-1} d(\zeta_i)$ et l'algorithme de Viterbi

([FORN73], [KUN89]) consiste à trouver le plus court chemin x_0^k à chaque position k et à avancer dans la chaîne jusqu'au point T . La mise en place de cet algorithme sera détaillée en (III).

I-3- Applications aux sommes littérales de chèques

Nous considérons la somme manuscrite d'un chèque comme étant une chaîne de Markov cachée de mots ou morceaux de mots. Les états du modèle seront donc ces mots. Les connaissances a priori (syntaxe) sur ces chaînes à vocabulaire limité liées à une base d'apprentissage conséquente vont nous permettre d'évaluer les 5 paramètres du HMM. Ce modèle va être utilisé pour la reconnaissance de phrase. Il pourra être facilement réappliqué aux lettres ou à toute autre structure décrivant un mot.

II- DÉTECTION DES MOTS

Le texte étant localisé sur l'image binarisée, il est nécessaire de détecter la ligne de lecture de la phrase et sa segmentation en mots.

Tout texte, qu'il soit imprimé ou manuscrit, présente des alternances de parties quasi-horizontales blanches et noires caractéristiques de la présence de lignes de mots. Quelles que soient les variations dues au scripteur, cette structure caractéristique est un des éléments fondamentaux de notre perception d'un texte écrit.

L'approche utilisée s'apparente aux approches descendantes (ou "top-down") utilisées pour déterminer la structure logique d'un document, qui procèdent depuis les éléments de niveau informatif le plus élevé (les blocs de texte) jusqu'à ceux de niveau inférieur recherchés (les mots). Les lignes de texte manuscrit peuvent être caractérisées par leurs lignes de base supérieures et inférieures ("base-line"). La localisation des mots est ensuite effectuée sur chacune des lignes.

II-1- Principe de localisation des lignes.

Le principe mis en place procède par analyse des profils de transitions qui présentent des valeurs maximums sur les zones de caractères sans extension. La détermination précise de ces zones permet de localiser les positions des corps de lignes en début, au milieu et en fin de texte. La valeur moyenne des profils détermine le seuil permettant de ne conserver que les zones de plus forte transition qui sont les corps des lignes de texte.

La hauteur moyenne des zones conservées sur l'ensemble des profils détermine la hauteur des caractères bas sur l'ensemble du texte ou encore hauteur moyenne des corps de lignes. La mise en correspondance des zones conservées des profils gauche et droit permet de déterminer les lignes de base recherchées qui sont les droites joignant la limite supérieure (inférieure) des zones conservées sur le profil gauche à la limite supérieure (inférieure) des zones conservées sur le profil droit. Finalement, on extrait les contours partiels des phrases qui s'appuient sur les lignes de base (figure 1).

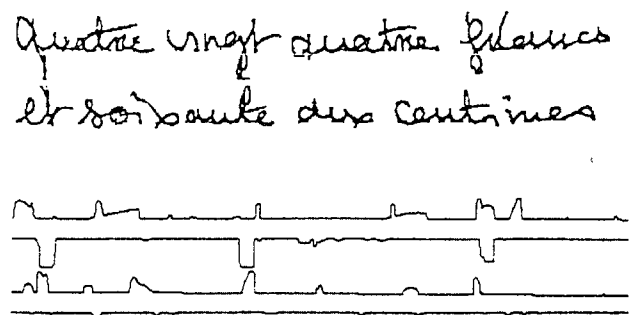


figure 1: Exemple de contours partiels

II-2- Segmentation en mots

La segmentation du texte en mots est une étape qui conditionne pour une large part la suite du traitement. En effet, la reconnaissance du texte se faisant mot à mot, il est indispensable de déterminer où ils commencent et où ils se terminent. De même, les extensions supérieures et inférieures sont des éléments caractéristiques très informantes sur le vocabulaire étudié. Ils seront utilisés comme caractéristiques de l'état "mot".

Dans le cas d'une écriture bien formée, on peut considérer que les espaces inter-mots sont de largeur bien supérieure à la largeur des espaces inter-lettres qui peuvent exister dans les mots manuscrits. On peut donc conclure dans ce cas à l'existence d'une répartition bi-modale des largeurs des espaces blancs dans le corps de ligne. Ceci est vérifié dans la pratique et représenté sur la figure 2 ci-dessous.

Toutefois, cette seule règle ne permet pas la prise en compte des fluctuations d'espacement des mots sur une écriture quelconque. Dans cette situation, plusieurs propositions de segmentation plausibles sont retenues sur les phrases qui seront analysées.

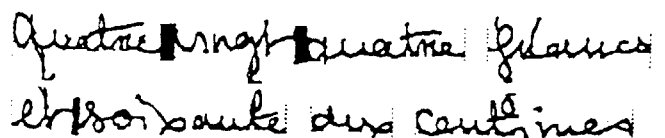


figure 2: Segmentation multiple d'une phrase, les séparateurs noirs indiquent une segmentation ambiguë.



III- MODULE DE RECONNAISSANCE

III-1- Le modèle

Le module de reconnaissance consiste à appliquer l'algorithme de Viterbi sur le modèle de Markov caché $\Lambda=(n,m,A,B,\Pi)$ suivant:

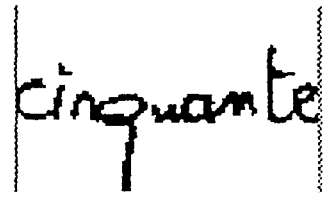
- $n = 27$ mots, c'est le nombre de mots du vocabulaire.
 - m nombre d'observations (relatif à la base d'apprentissage).
 - A matrice de transitions obtenue sur une base théorique de montants de chèques.
 - B matrice des probabilités d'observations (relative à la base d'apprentissage).
 - Π vecteur des probabilités des états initiaux.
- A et Π sont obtenus de la manière suivante par simulation:
- on génère des montants séquentiels.
 - on vérifie qu'il sont corrects syntaxiquement.
 - on comptabilise les différentes transitions de mot.

On obtient alors après normalisation, des probabilités d'enchaînement de mots.

m et B sont obtenus par apprentissage sur une base de montants étiquetés après codage des états "mots".

III-2- Codage des états "mots"

Notre soucis a été de simplifier la définition d'un état. A partir des contours partiels des mots obtenus dans (II-1), on repère hampes et jambages du mot, ce qui permet un codage simple de l'état. En effet, dans le cas d'une écriture bien formée, les hampes et les jambages sont des extensions suffisamment importantes pour qu'on les distingue facilement en observant la répartition des valeurs des extensions supérieures et inférieures. Les primitives extraites sur les mots sont ainsi de trois types: HAMPE, JAMBAGE, RIEN.(voir exemple figure 3)



Observations : RIEN - JAMBAGE - RIEN - HAMPE
figure 3 exemple de codage pour le mot cinquante

III-3- Applications de l'algorithme de Viterbi

A l'observation $n^o=1$ on va associer une liste d'états possibles avec une probabilité associée : c'est la matrice B des $b_i(l)$ pour $i=1, \dots, n$, $l=1, \dots, m$.

m est le nombre d'observations obtenu à l'issue de la phase d'apprentissage sur une base étiquetée.

L'étape de reconnaissance consiste alors à construire les différentes chaînes possibles et à ne garder que les chaînes correspondant aux parcours les plus courts dans le treillis d'états. L'exemple des figures 4 et 5 illustre le fonctionnement de l'algorithme. Si on prend le chemin le plus court (première ligne), et si on utilise la base présentée en (IV):

0,14 est un élément de la matrice Π_j

0,27 et 0,15 sont des éléments de la matrice A

0,13 et 0,57 sont des éléments de la matrice B

La phrase retenue est : *quatre vingt francs*

$$d(x_0^k) = 8,25$$

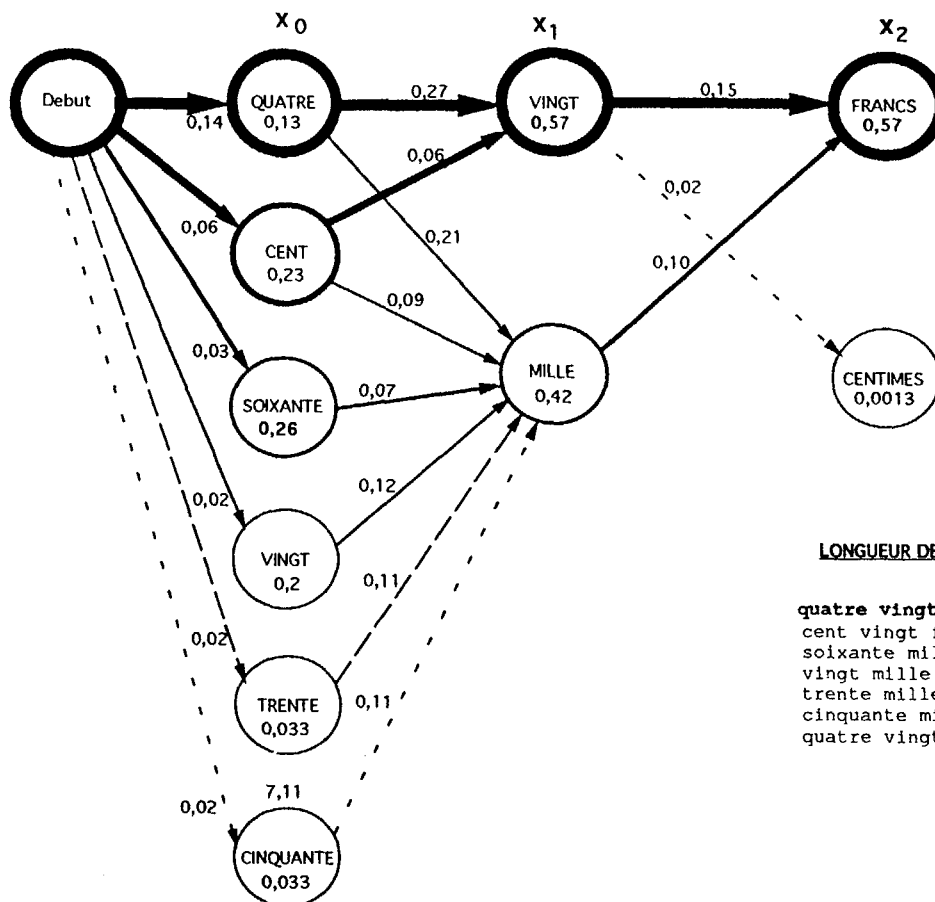


figure 4 : Exemple de construction de chaîne et choix pour la solution du parcours le plus court.

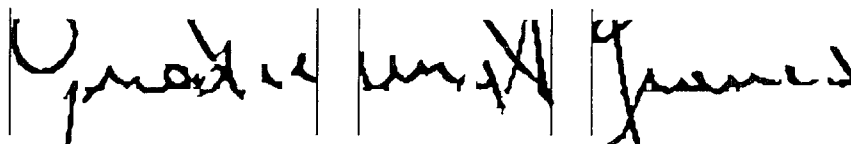


Figure 5 : Image du chèque de l'exemple figure 4

taux de reconnu en position					au delà du 5 ^{ieme}
1	2	3	4	5	
2%	3%	2%	2%	3%	88%

Figure 6a : Taux de reconnaissance du classifieur mot.

reconnu dans la phrase en position					au delà de
1	2	3	4	5	la cinquième position
61%	6%	3%	1%	3%	26%

Figure 6b : Taux de reconnaissance de mot du classifieur phrase après module de Viterbi.

IV- RÉSULTATS

IV-1- Le Support

La base mise à notre disposition contient environ 1000 montants de chèques, ce qui correspond à près de 3000 mots. Cette base est formée d'éléments provenant du Service de Recherche Technique de la Poste (SRTP), à Nantes, et de la société Matra MS2i, à Saint Quentin en Yvelines.

Ces sommes sont particulièrement dégradées, du fait de la mauvaise qualité d'écriture et des problèmes d'élimination des fonds de chèque (double barre, superposition de ligne : voir figure 5). Cette base est segmentée et étiquetée en mots.

L'apprentissage du classifieur mot est effectué sur cette base.

IV-2- Les Résultats sur la base

Une première étape a consisté à évaluer les taux de reconnaissance du classifieur mot, ceci afin d'avoir un critère de comparaison pour le module de reconnaissance de sommes de Viterbi.

La classification, au niveau des mots est tout d'abord effectuée par l'extraction de primitives très simples (Hampe, Jambage, Rien), ce qui explique le taux de classification médiocre (moins de 15% dans les cinq premières propositions) (figure 6a).

La construction des chaînes de Markov cachée sur les états (mots) permet de remonter les taux de reconnaissances à 60-70% (dans les cinq premières phrases) comme il est indiqué en figure 6b.

IV-3- Analyse des résultats

Nous remarquons immédiatement (figure 6a) les résultats médiocres du classifieur "mot". En effet, le codage simplifié ne permet pas de différencier certains cas simples. Par exemple, le codage du *cinquante* (figure 3) pourrait également correspondre au codage d'un *sept* (puisque aucune notion de longueur n'est introduite). Mais lorsque nous construisons les chaînes de Markov de mots, la différence des taux de reconnaissance entre ces deux mots devient grande (figure 6b). Ainsi, si l'état précédent est correct, la bonne réponse sera privilégiée.

Cette méthode ne permet évidemment pas de lever tous les doutes. Par exemple, les mots *deux* et *trois* peuvent être morphologiquement et syntaxiquement identiques. Nous ne pouvons alors conclure. Un affinement du codage des états ainsi

que des HMM sur ces états devraient différencier le *deux* du *trois*.

CONCLUSION ET PERSPECTIVES

Comme nous venons de le souligner, la très grande simplicité des états et donc la souplesse du module présenté ne peut suffire à identifier clairement les chaînes. Des paramètres morphologiques peuvent être encore ajoutés, comme la longueur du mot.

De même, la construction des chaînes de Markov cachées d'ordre supérieur (2 voire 3), devrait améliorer le taux de reconnaissance.

De plus, la syntaxe connue a priori sur l'enchaînement des lettres dans le mot doit intervenir dans le module de reconnaissance, même si l'on sait que la segmentation en lettres d'un mot manuscrit reste du domaine de l'illusion. Ces lettres ou tout autre état remplaceront les "mots" dans le module présenté, la "phrase" étant alors remplacée par le "mot".

Remerciements

Nous remercions le SRTP ainsi que la société MS2i de nous avoir fourni une partie de leur base pour tester nos algorithmes.

Références:

[COURT92] Courtellemont P., Olivier C., Avila M., 2-D Modeling and Hough Transform for document analysis, IEEE-IES, IECON92 San-Diego CA (USA), vol 3, pp1325-1330, 1992.

[FORN73] David Forney G., The Viterbi Algorithm, Proceeding of IEEE, vol 61, n°3, pp 268-278, 1973.

[KUN89] Kundu A., He Y., Bahl P., Recognition of Handwritten word: first and second order hidden Markov model based approach, Pattern Recognition, vol 22, n°3, pp 268-278, 1992.

[PAQ93] Paquet T., Lecourtier Y., Recognition of handwritten sentences using a restricted lexicon, Pattern Recognition, vol 26, n°3, pp 391-407, 1993.

[RAB89] Rabiner L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceeding of IEEE, vol 77, n°2, pp 257-286, 1989.